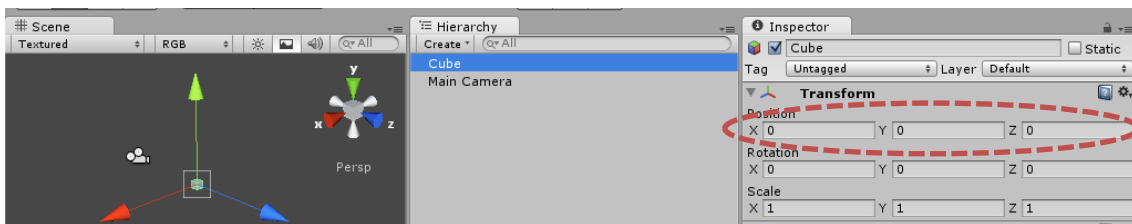
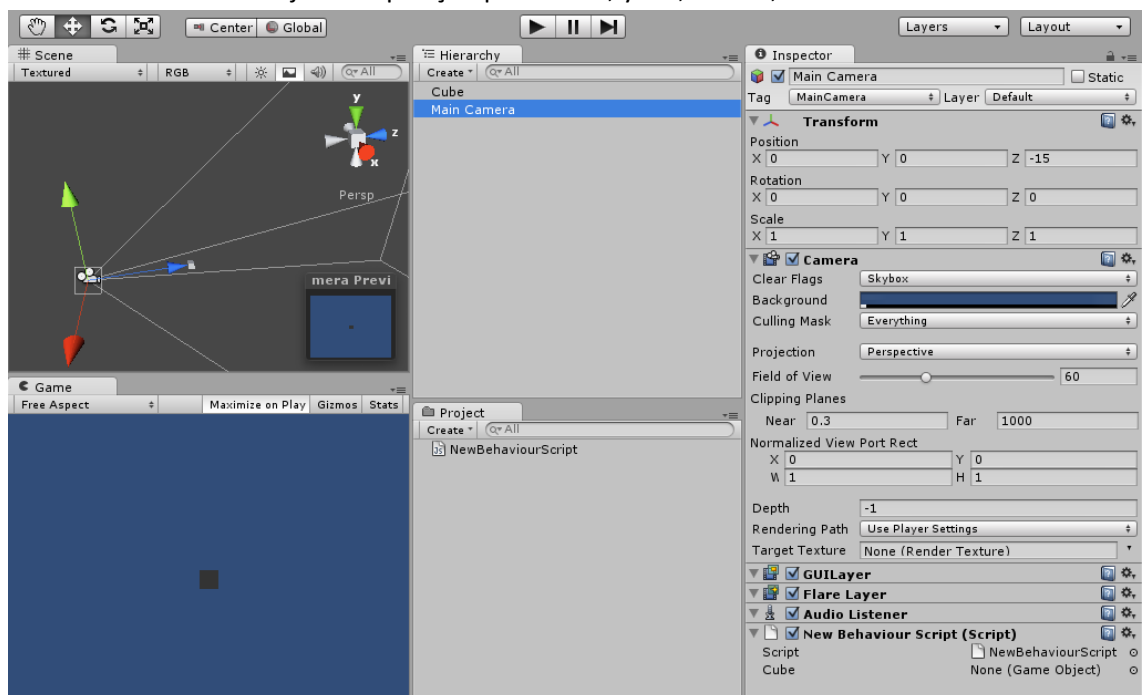


- 1) Tutorial Space Invaders Parte: 01 (referente a aula do dia 02/12/2010)
- 2) Objetivo: Aprender a movimentar objetos no espaço 3d.
- 3) Com o Unity aberto:
  - a) Inicie um projeto novo.
- 4) Preparar a cena básica (neste momento):
  - a) Adicione um cubo (será a nave inimiga que iremos movimentar):
    - i) GameObject → Create Other → Cube
  - b) Ajustar a posição do cubo: com o cubo selecionado:
    - i) Na janela *Inspector* ajustar os valores position na aba Transform:
    - ii) Ajustar manualmente na janela os valores:  $x = 0$ ;  $y = 0$ ;  $z = 0$ ;

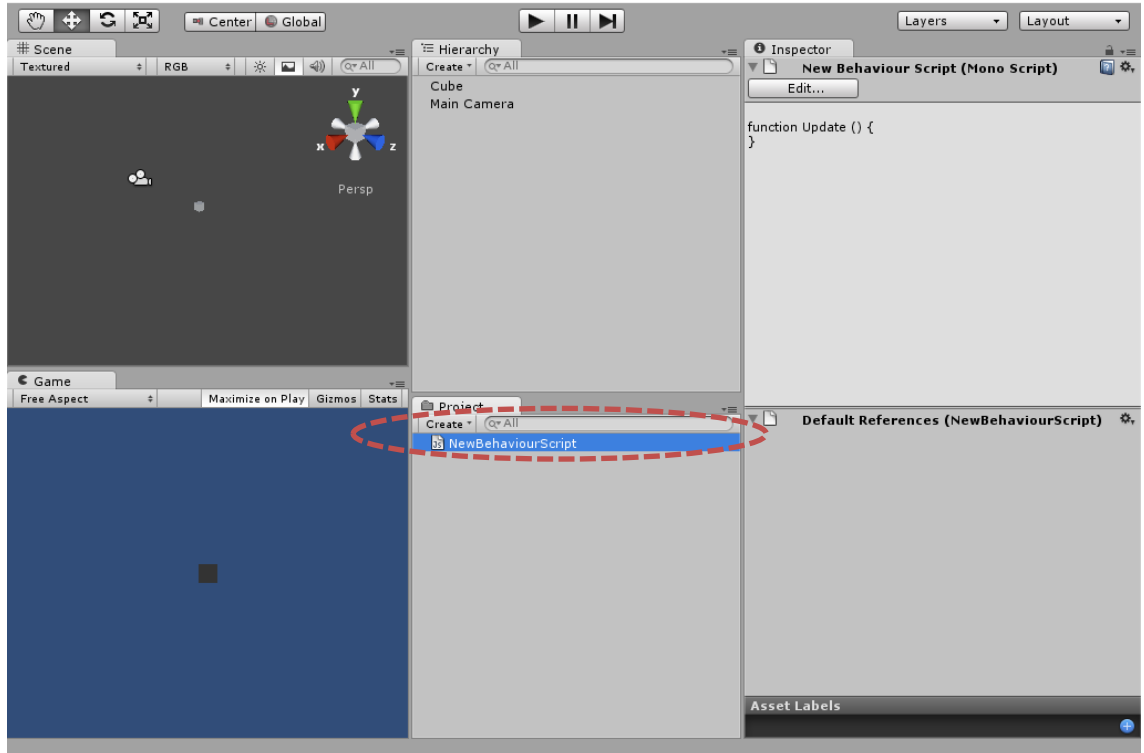


- 5) Ajustar a posição da câmera para que possa visualizar o cubo (posição temporária):
  - a) Selecionar a câmera e ajustar a posição para:  $x = 0$ ;  $y = 0$ ;  $z = -15$ ;

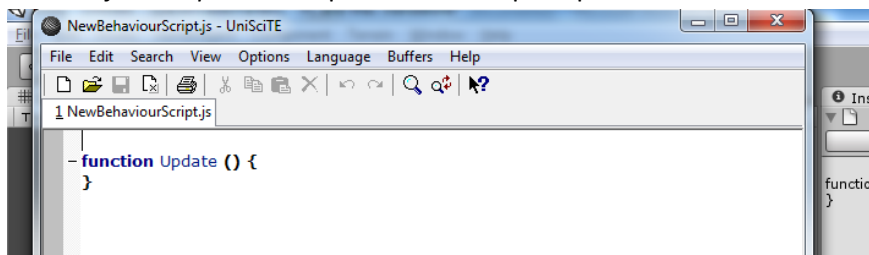


- b)
  - i) Observação: caso a organização do layout de janelas não esteja igual ao da figura acima é possível mudar isso em: Windows → Layouts → Unity 2 (caso esteja utilizando o Unity 3).
- 6) Iniciar a programação do *javascript* que controlará o objeto *Cube*:
  - a) Assets → Create → Javascriptaça

b) Observar que um novo arquivo será criado na janela Project:

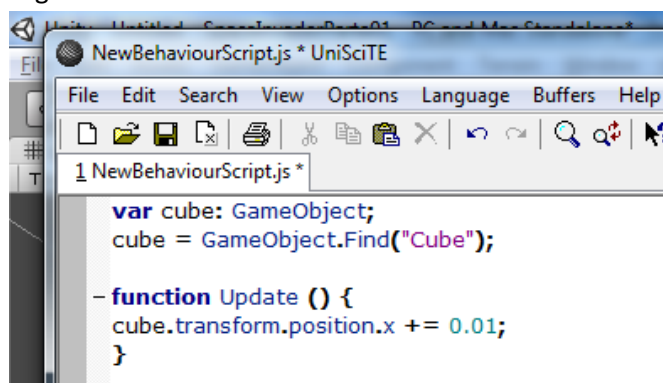


c) Editar o *javascript*: dois cliques sobre o arquivo para abrir o editor:



d)  
e) Testar o script com um código simples:

i) Digitar as linhas:

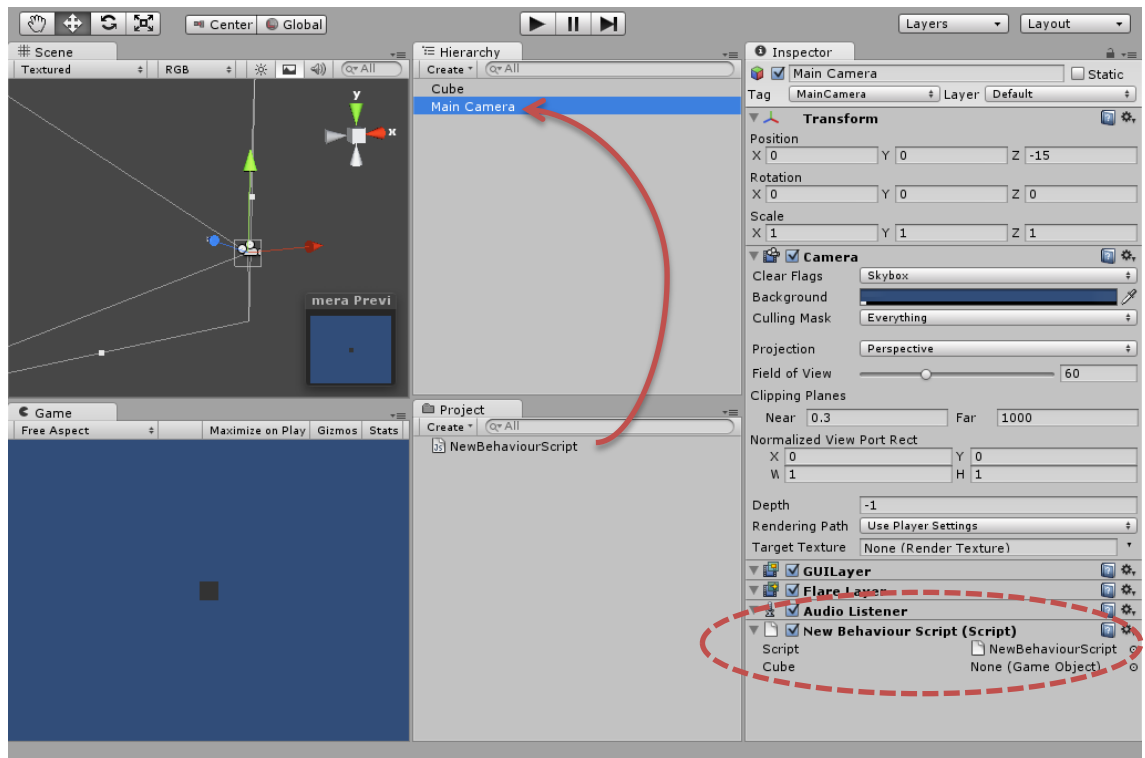


ii)

7) Associar o *javascript* com algum objeto da cena:

a) Arraste o javascript NewBehaveiourScript.js com a MainCamera

- b) O *javascript* é adicionado ao objeto camera como um novo componente na janela Inspector:



- a) Executar o jogo: basta clicar no botão “play”;  
 i) O resultado será o movimento do cube para direita;  
 2) Programar o movimento de vai e vem no eixo x:  
 a) Acrescentar as linhas:

```

1 NewBehaviourScript.js
var cube: GameObject;
cube = GameObject.Find("Cube");
var vx; //velocidade na direção x
vx = 4;

var sentido; //variável de controle que informa se a nave está indo para direita ou esquerda.

sentido = 1;

- function Update () {

    //Mover em x a todo instante com a velocidade vx com sinal dado pela variável sentido
    cube.transform.position.x += vx*sentido*Time.deltaTime;

    //Inverter o sentido quando chegar na lateral direita:
-   if (cube.transform.position.x >= 8 & sentido == 1){
        ...
        sentido = -1;
    }

    //Inverter o sentido quando chegar na lateral esquerda:
-   if (cube.transform.position.x <= -8 && sentido == -1){
        ...
        sentido = 1;
    }

}

```

- b) Após salvar esse *javascript* execute o jogo: o cube moverá para direita e esquerda.  
 3) Programar o movimento vertical no eixo y:  
 a) Toda vez que atingir as laterais o cube movimentará dy unidades no eixo y (vertical):

b) Acrescentar as linhas:

```
NewBehaviourScript.js *
var cube: GameObject; //declarar um objeto da classe GameObject
cube = GameObject.Find("Cube"); //associar a variável cube ao modelo com nome Cube

var vx; //velocidade na direção x
vx = 4;

var dy; //deslocamento na vertical em y
dy = 0.5;

var sentido; //variável de controle que informa se a nave está indo para direita ou esquerda.
sentido = 1;

function Update () {
    //Mover em x a todo instante com a velocidade vx com sinal dado pela variável sentido
    cube.transform.position.x += vx*sentido*Time.deltaTime;
    //Inverter o sentido quando chegar na lateral direita:
    if (cube.transform.position.x >= 8 & sentido == 1){
        sentido = -1;
        descer();
    }
    //Inverter o sentido quando chegar na lateral esquerda:
    if (cube.transform.position.x <= -8 && sentido == -1){
        sentido = 1;
        descer();
    }
}

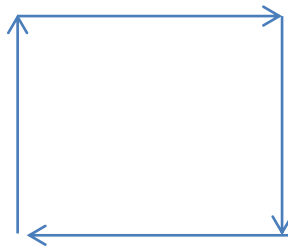
function descer() {
    cube.transform.position.y -= dy;
}
}
```

4) Fim da primeira parte.

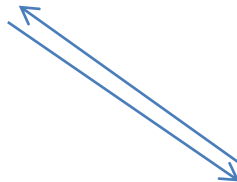
5) Desafio para verificação da aprendizagem:

a) Elabore novos movimentos para o objeto cube seguindo os digramas:

(1) Quadrado:



(2) Diagonal:



Prof. Rafael João Ribeiro IFPR – Campus Telêmaco Borba.

Curso Técnico em programação de Jogos Digitais – Uso exclusivo para os alunos do curso.